

REMARKS/ARGUMENTS

The Examiner is thanked for granting the Applicant a telephonic-interview on May 3, 2005. As was proposed during the interview, claim 1 has been amended to additionally recite features recited in claims 5, 6, 7 and 8. The new claims 22-35 recite similar features as those recited in claim 1. Substance of the telephonic-interview is summarized below for the Examiner's convenience.

In the Office Action, the Examiner has rejected claims 1, 4-12, 15 and 17-19 and 21 under 35 U.S.C. 103(a) as being unpatentable over U.S. patent No. 6,738,977 (*Berry et al.*) in view of the U.S. patent No. 6,339,841 (*Merrick et al.*). This rejection is fully traversed below.

The claimed invention

As noted in the specification, a method information portion (210) can be arranged to provide information relating to one or more methods. These methods can be, for example, be implemented as a table similar to a table of method information implemented in a standard Java class file. In addition to the method information portion (210), an internal class representation (206) includes a method reference portion (220) associated with the methods contained within the internal class representation. The method reference portion (220) can be arranged to include any reference cell associated with the class. By way of example, the reference cells may be linked together using a link-list construct in accordance with one embodiment of the invention. Each reference cell can include information that is useful in invoking a method (specifications, pages 8-10). As also noted in the specification, a reference cell (400) can include a method name field (402), a method signature field (404), and a code reference field (406). Conventionally, the signature is constructed into a form usable by the virtual machine using a series of calls to the constant pool at runtime when the method is invoked. Although this works well, it is relatively slow. One advantage to including the signature in the reference cell is that a signature suitable for direct use by the virtual machine can be constructed during loading and either stored directly in the reference cell or stored in a location that is referenced by the reference cell (specifications, pages 8-10).

Rejection of the claims

In the Office Action, the Examiner has noted that *Merrick et al.* does NOT show a method reference portion that includes one or more reference cells, but this teaching is inherent of a method table (Office Action, page 8). It is respectfully submitted that the fact that a certain result or characteristic may occur or be present in the prior art is NOT sufficient to establish the inherency of the result or characteristic (see, for example, MPEP 2112). In this case, a method reference portion is clearly NOT necessarily present in *Merrick et al.*

Moreover, it is respectfully submitted that *Merrick et al.* does NOT teach or even remotely suggest: a method reference portion comprising a list, representing all of a plurality of selected methods, which is arranged as a sequence of reference cells each representing one of the plurality of the selected methods, wherein each of said reference cells includes a method name field, a method signature field, and a method code field. It is noted that *Merrick et al.* teaches a conventional method table, but clearly *Merrick et al.* does NOT teach this feature.

Further, it is respectfully submitted that *Merrick et al.* does not teach or suggest selecting information from a class file in a memory portion after the class file has been loaded in its entirety into the memory portion. It is noted that *Merrick et al.* relates to a class loading model. However, *Merrick et al.* teaches performing a post compilation process on object code to generate component object code, namely, Meta data and methods. The post compilation process is applied in order to break down the self-contained structure of a class and convert a self-contained class into individually accessible components of the class (*Merrick et al.*, Col. 3, lines 36-45). As such, it is respectfully submitted that *Merrick et al.* does not teach selecting information from the class file. Instead, *Merrick et al.* teaches: first breaking down the object code into individually accessible components and then loading these individual components into the virtual machine (*Merrick et al.*, Col. 3, lines 36-43).

Conclusion

Based on the foregoing, it is submitted that all pending claims are patentably distinct over the cited art of record. Additional limitations recited in the independent claims or the dependent claims are not further discussed because the limitations discussed above are sufficient to distinguish the claimed invention from the cited art. Accordingly, Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 500388 (Order No. SUN1P815). Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP



R. Mahboubian
Reg. No. 44,890

P.O. Box 70250
Oakland, CA 94612-0250
(650) 961-8300